# Cray XC System Diagnosability Roadmap

Jeffrey J. Schutkoske
Platform Services Group (PSG)
Cray, Inc.
St. Paul, MN, USA
jjs@cray.com

*Abstract*— **This paper highlights the current capabilities and the technical direction of Cray XC System level diagnosability. Cray has made a number of enhancements to existing diagnostics, commands, and utilities as well as providing new diagnostics, commands and utilities. This paper reviews the new capabilities that are available now, such as the Simple Event Correlator (SEC), Workload Test Suite (WTS), Node diagnostics and HSS diagnostic utilities. It will also look at what is planned for in the upcoming releases, including the initial integration of new technologies from the OpenStack projects.**

***Keywords-Cray XC, diagnostic, diagnosability, computer architecture***

## I. INTRODUCTION

System Diagnosability is a suite of software tools designed to provide Cray field support and end customers a tool chain to quickly and reliably identify hardware and software problems in the Cray XC system.

The Hardware Supervisory System (HSS) system management platform supports a variety of hardware platforms with varying management and control capabilities. For this reason, HSS system management provides abstracted interfaces for important hardware access operations and allow the development of customized drivers, where necessary, which interface directly with the hardware. There are a number of basic infrastructure capabilities that enable higher-level functions within the HSS system management environment including system discovery and inventory management, node management, high-speed network (HSN) management, and system infrastructure management. The higher-level functions include monitoring, logging, notification, resiliency, power and cooling management, diagnostics, and at-scale debug.

HSS provides a highly scalable management infrastructure allowing a system from a single blade to thousands of blades to be controlled and monitored from a single system management node. It features a relatively light footprint in terms of hardware and software, and performs monitoring and management out-of-band, so impact on user applications on Cray Linux Environment (CLE) is minimal, with no significant jitter introduced on the compute node. It also uses standard Linux images, with some additional drivers where needed, and a standard Gigabit Ethernet network.

System Diagnosability is not just about diagnostics. Diagnostics are just one aspect of the tool chain that includes BIOS as well as user commands. It includes power, thermal, and sensor data, and event logs. There are component level tests that are used to checkout the individual components, but quite often issues do not appear until substantial scale (20+ cabinets) is reached. From experience over the last few years, we have seen that no single tool or diagnostic can be used to identify problems, but rather multiple tools and multiple sources of data must be analyzed to provide proper identification and isolation of hardware and software problems.

System Diagnosability of Cray XC system components include the Aries High Speed Network (HSN), Intel compute processors, Intel Xeon PHI co-processors, Nvidia GPUs, Memory, Intel Quick Path Interconnect (QPI), Peripheral Component Interconnect Express (PCIe), Cray cabinet power and cooling, as well as, storage solutions and connectivity to storage solutions.

Cray has made a number of enhancements to existing diagnostics, commands, and utilities as well as providing new diagnostics, commands, and utilities. These enhancements include:

- Node Stress and Performance Diagnostics
- Intel Haswell Diagnosability
- Nvidia GPU Diagnosability for CUDA 6.5
- I/O Diagnosability
- Workload Test Suite (WTS)
- HSS Diagnostic Utilities
- HSS Power and Cooling Diagnosability
- HSS Telemetry Data
- HSS Controller Monitoring
- System Notification
- Cray Linux Environment (CLE) Enhancements

This paper reviews these new capabilities in Sections II - XII. In Section XII it looks at what is planned for in the upcoming releases, including the initial integration of new technologies from the OpenStack projects.

## II. NODE DIAGNOSABILTY ENHANCEMENTS

There have been a couple of enhancements to improve node diagnosability as follows:

- Node Level Stress (NLS) diagnostic
- Node DGEMM performance diagnostic

## *A. Node Level Stress (NLS) Diagnostic*

The Node Level Stress (NLS) diagnostic, *xtnls*, is a new diagnostic that is a collection of diagnostic programs and libraries. The libraries provide multiple classes of test algorithms along with node resource management. Each program utilizes the library to integrate the resource management framework and test algorithms to create a diagnostic intended to validate functionality of specific node resources.

The CPU instruction test selects a set of random CPU instructions to execute, executes the set of instructions, and validates the results.

The NUMA memory test executes and validates random NUMA interleaving as well as local and remote memory tests.

The NLS diagnostic is designed to ride through and recover from non-fatal hardware errors. The parent thread creates a monitoring thread to control and track the progress of each diagnostic thread group executing test algorithms. The monitoring thread cleans up the diagnostic thread group if the executing test algorithm stalls. The parent thread also monitors progress of monitoring threads and forces a clean up should the whole hierarchy of threads stall. Prior to any clean up, the thread status is accumulated to provide stall state for later potential diagnosis.

The NLS diagnostic failure output identifies the failing node as well as the expected and actual test results as shown below.

```
c0-0c0s15n3 nid00063 id=7, cpu=0, core=8, Mandelbrot
c0-0c0s15n3 nid00063 id=2, cpu=0, core=2, Mandelbrot
c0-0c0s15n3 nid00063
        FE_UPWARD ADDS (null) OP (1073741824)
        Expected 0x00000000 Actual 0xaaab4a29
c0-0c0s15n3 nid00063 id=6, cpu=0, core=6,
        Basic x86_64+BMI1+BMI2
c0-0c0s15n3 nid00063 id=13, cpu=0, core=14,
        Basic x86_64+BMI1+BMI2
c0-0c0s15n3 nid00063 id=8, cpu=0, core=9,
        Basic x86_64+BMI1+BMI2
c0-0c0s15n3 nid00063 id=3, cpu=0, core=3,
        SSE Double Precision
```

Figure 1. NLS Diagnostic Failure Output

## *B. Xeon Processor Performance*

The *xtcpudgemm* diagnostic has been enhanced for greater control over testing each Intel processor core. In the past the diagnostic tested the dual socket node as an entity. It has also been enhanced to utilize the Intel AVX2 instruction set to increase the processor utilization.

When the *xtcpudgemm* diagnostic detects a performance issue, the diagnostic prints the actual value and the expected value. The bin value is the delta from the normal clock in MHz. The expected bin value is greater than or equal to -450 for Intel Haswell processors.

```
c0-0c2s9n2, nid00166, Fail:
        Bin actual: -455.14,
        Bin expected greater than or equal to: -450
c0-0c2s9n2, nid00166, Fail
c0-0c2s9n2, nid00166, 1 failure(s)
```

Figure 2. Processor Diagnostic Performance Failure Output

In the above example, the processor is underperforming relative to the expected variation from the expected clock. These diagnostic enhancements have allowed the diagnostic to isolate down to the failing core within the Intel processor.

## III. Intel Haswell Diagnosabilty Enhancements

The Cray HSS, BIOS, and diagnostics have all been enhanced to support the Haswell processor in the Cray XC system. Most changes are transparent to the system administrator. There are a couple of changes that are unique to the Intel Haswell processor.

The Intel Machine Check Architecture (MCA) decode is a superset of Intel SandyBridge and Intel IvyBridge. The *xtmcadecode* utility handles all 3 Intel processor types plus the Intel Phi Knights Corner (KNC) co-processor. The DIMM numbering scheme is the same as Intel SandyBridge and Intel IvyBridge processors. Banks 0-8 are basically the same. Banks 9-16 are still memory controllers, but are named slightly different. The remaining banks are different for Intel Haswell.

Identifying the failing DIMMs and CPUs remains the same.

To print a summary with error counts in the log file:

*xthwerrlog --count -f hwerrlog_file*

To print all DIMM errors with location info:

*xthwerrlog --dimminfo -f hwerrlog_file*

The Intel Haswell PECI support was enhanced. HSS has included this functionality of the monitoring of the Intel Haswell processor as follows:

- CPUID, Platform ID, and Power Controller Unit ID.
- Machine Check error log (keeps history of errors).
- Access to CPU Turbo Ratio.
- On-board Voltage Regulator debug data.
- Temperatures: CPU die temps, thermal margin temps, operating target temps, PCH temp, Voltage Regulator temps, DIMM temps, and inlet air temp.
- Thermal event status.
- Power Management: Power throttle duration, current draw limits, DDR power limit, node power consumption, power supply status, and Thermal Design Point (TDP) high/low/ratio.

## IV. NVIDIA GPU DIAGNOSABILTY ENHANCEMENTS

Nvidia has upgraded the CUDA software stack to version 6.5. The Cray GPU on-line diagnostics were designed and developed using CUDA 6.5. The Cray GPU diagnostics that were upgraded to support CUDA 6.5 are as follows:

- **nvidia-healthmon**: Nvidia provides a health monitor tool, *nvidia-healthmon*, facilitates running a basic functional test on Tesla GPUs between cluster jobs [1].
- **xkmemtest**: The Cray GPU Memory test, *xkmemtest*, tests GPU memory for hardware errors and soft errors using CUDA.
- **xkdgemm**: The Cray DGEMM test, *xkdgemm*, is a standard double precision floating point matrix multiply application. It utilizes CUDA to execute the matrix multiply on the Nvidia GPU and requires the standard CUDA BLAS library.
- **xkbandwidth**: The Cray PCIe bandwidth test, *xkbandwidth*, is used to measure the memcopy bandwidth of the GPU.
- **xkstress**: The Cray GPU stress test, *xkstress*, performs stress and performance tests across nodes. This test includes streams, GEMM, and PCIe tests. It is an MPI application that compares performance results across the blades, cabinets, and system.
- **xkcheck**: The Cray GPU check application, *xkcheck*, validates the Nvidia GPU hardware configuration, firmware version, CUDA Driver API, NVML API and CUDA runtime comparisons. The application can display the Node IDs and Cray name (cname) for each delta found. It is an MPI application that compares the configuration results across the blades, cabinets, and system.

## V. I/O DIAGNOSABILITY ENHANCEMENTS

There have been a couple of enhancements for I/O diagnosability. The first enhancement was made to the HSS diagnostic utility, *xtcheckhss,* to validate the PCIe I/O cards in the system. The second enhancement is including the disk diagnostic, *xtxdd*, to validate the attached disks.

### *A. HSS diagnostic utility (xtcheckhss)*

The HSS diagnostic utility, *xtcheckhss*, is executed on the SMW. It provides a validation of the PCIe I/O cards that are connected to the Cray I/O Base Blade (IBB).

There are a number of errors that are detected as follows:

1. Speed mismatch (i.e. trained to Gen1 rather than Gen2)
2. Width mismatch (i.e. trained to x4 rather than x8)
3. Missing PCIe card (i.e. card wasn't detected by BIOS, but reported by IBB Microcontroller

The nodes on the IBB must first be bounced before executing the HSS diagnostic utility. The data provided includes the Cray name (cname), slot, description, target speed, trained speed, target width, and trained width.

```
c1-0c0s0n1  2
            Mellanox_ConnectX_IB_QDR_MT26428
            Gen2     Gen2       x8       x8
c1-0c0s0n1  3
            Empty
c1-0c1s3n1  2
            Intel_Gigabit_ET2_Quad_Port_Server_Adapter
            Gen1     Gen1       x4       x4
c1-0c1s3n1  3
            QLogic_ISP2532_8Gb_Fibre_Channel_HBA
            Gen2     Gen2       x4       x4
c1-0c1s3n2  0
            Intel_Gigabit_ET2_Quad_Port_Server_Adapter
            Gen1     Gen1       x4       x4
c1-0c1s3n2  1
            QLogic_ISP2532_8Gb_Fibre_Channel_HBA
            Gen2     Gen2       x4       x4
c1-0c1s5n1  2
            Mellanox_MT27500_ConnectX-3
            Gen3     Gen3       x8       x8
```

Figure 3. HSS Diagnostic I/O Device Output

### *B. Disk I/O Test (xtxdd)*

Each PCIe disk device vendor card and vendor device driver that is installed in the Cray XC system is tested using eXtreme DD (XDD) [2], under the Cray Linux Environment (CLE). Cray configures *xtxdd* to perform data transfer operations between memory and multiple disk devices or files and to collect performance information about the I/O operations. *xtxdd* creates multiple threads, one for each device or file under test. Each created thread performs a number of setup operations and enters a serialization barrier. All threads start executing when the last thread completes its initialization and enters the serialization barrier. As threads complete, they enter a pass barrier, so that all threads operate concurrently. The threads are assigned to specific processors.

*xtxdd* has a number of command line options that allow the user to effectively control the testing. Some examples are as follows:

- **Datapattern**: Can be random, sequenced, ascii or hex.
- **mbytes**: Specifies the integer number of megabytes to transfer on each pass.
- **Passes**: Number of passes to perform.
- **Queuedepth**: Specifies the number of commands to send to each target at one time.
- **Randomize**: Causes the seek locations to be randomized between passes.

The data read from the device is compared with the specified data pattern. The performance data can also be manually compared between runs of *xtxdd*. Changes in performance can be an early indication of other problems.

## VI. WORKLOAD TEST SUITE (WTS)

The new Workload Test Suite (WTS) consists of a control script, *xtsystest*, and a number of benchmarks and diagnostics. The benchmarks and diagnostic tests are used to simulate a generic application workload to verify that the system is ready to execute user applications. In some cases a customer may have one or two applications that are representative of the workload on-site. In other cases standard benchmarks are used as follows:

1. **Intel MPI Benchmarks (IMB):** Performs a set of MPI performance measurements for point-to-point and global communication operations for a range of message sizes [3].
2. **High Performance Computing Challenge (HPCC):** Consists of 7 test including HPL, DGEMM, STREAM, PTRANS, Random Access, FFT, and Communication bandwidth and latency [4].
3. **High Performance Linpack (HPL):** Cray uses HPL as two different tests: one to test out the processor running HPL (DGEMM) and the second to use as large of memory foot print as possible [5].

These applications are pre-configured and pre-compiled versions of the applications that a customer site may run on a regular basis.

Baseline expectations of performance will vary depending on a number of system specific settings, the number of tests that are included in the suite of tests, and the number of resources that are targeted for each test in the suite. One particular benchmark that needs to be handled with care is HPL. A single copy of *xhpl* run across cabinets can take days to run. The goal is to have *xhpl* run for around one hour.

```
checking results: xtimb_check
  check_logs: total error count: 2
  checking results: xtmemtest_check
  2014-09-03 13:43:53,126 - xtmemtest_test - INFO - check_logs: no
problems found
  checking results: xthpcc_check
  check_logs: total error count: 2
  checking results: xkdgemm_check
  2014-09-03 13:43:53,375 - xkdgemm_test - INFO - check_logs: no
problems found
  checking results: xtphidgemm_check
  2014-09-03 13:43:53,503 - xtphidgemm_test - INFO - check_logs: no
problems found
  checking results: xkstress_check
  2014-09-03 13:43:53,627 - xkstress_test - INFO - check_logs: no
problems found
  checking results: xthpl_check
  2014-09-03 13:43:53,764 - xthpl_test - INFO - check_logs: no problems
found
  checking results: xtonline_diags_check
  2014-09-03 13:43:53,906 - xtonline_diags_test - INFO - check_logs: no
problems found
  checking results: xkmemtest_check
  check_logs: total error count: 4
  checking results: xkbandwidth_check
  2014-09-03 13:43:54,199 - xkbandwidth_test - INFO - check_logs: no
problems found
  checking results: xtphibandwidth_check
  2014-09-03 13:43:54,349 - xtphibandwidth_test - INFO - check_logs: no
problems found
  checking results: xtphighpl_check
  2014-09-03 13:43:54,508 - xtphighpl_test - INFO - check_logs: no
problems found
  checking results: xkcheck_check
  2014-09-03 13:43:54,657 - xkcheck_test - INFO - check_logs: no
problems found
  checking results: xtcpudgemm_check
  2014-09-03 13:43:54,827 - xtcpudgemm_test - INFO - check_logs: no
problems found
```

Figure 4. Workload Test Suite Failure Output

The above output from the control script, *xtsystest*, shows that *xtimb_test*, *xthpcc_test,* and *xkmemtest_test* have reported errors.

## VII. HSS Diagnostic Utilities

The HSS diagnostic utility, *xtstresshss*, executes a processor and a memory diagnostic on each HSS blade and cabinet processor (Tolapai). The utility executes on the SMW and copies the tests to the HSS blade or cabinet processor under test.

The utility checks for any errors encountered by the HSS blade or cabinet processors while running the tests. The types of errors checked for include; controller reboots, MCEs, and other error messages in the /var/log/messages file on the HSS blade or cabinet processor.

The HSS stress diagnostic utility uses the controller boot script to check for unexpected errors on the HSS blade or cabinet processor. It is defaulted to execute on each blade and cabinet processor for 20 minutes.

A DRAM failure on the blade or cabinet processor results in error messages as shown below.

```
c0-0c1s2 Tolapai error register GLOBAL_FERR: 0x8000000

c0-0c1s2 Tolapai error register DRAM_FERR: 0x40
```

Figure 5. Controller DRAM Failure Output

A Watch Dog Timeout failure on the blade or cabinet processor results in error messages as shown below.

```
c0-0c3s7 Tolapai watchdog timeout
```

Figure 6. Controller Watchdog Timeout Failure

This is a stress test and should not be executed when Cray Linux Environment (CLE) is booting.

## VIII. HSS Power and Cooling Diagnosability

The Cascade Cabinet (CC) controller has enhanced validation checks that are performed at initial system power on and boot times. These checks include:

- Emergency cabinet power down
- Cabinet cooling alert
- Cabinet power alert

- Cabinet air temperature alert
- Cabinet water pressure alert
- Cabinet water leak alert
- Cabinet blower alert
- Cabinet rectifier alert

When these alerts are triggered, a full cabinet power cycle is required after the repair action has been taken. The HSS diagnostic utility, *xtcheckhss,* is used to trouble-shoot any cabinet power or cooling issue.

The *xtcheckhss* (*xtcheckhss –health*) cabinet power and cooling messages include:

- CC Microcontroller initialization phase errors
- CC Microcontroller power on phase errors
- CC Microcontroller run-time errors

These alerts are monitored by specific SEC rules.

## IX. HSS Telemetry Data

The System Environment Data Collections (SEDC) software monitors the system health and records the environmental data coming from the system's hardware components. Collecting real-time data from the components helps to improve maintenance of the hardware, provides monitoring of hardware, and provides detailed data for analysis of failures.

SEDC stores the sensor data in automatically rotated flat files on the SMW in the location specified in sedc_srv.ini configuration file. These flat files can get quite large, that makes search for individual sensor's value time consuming and difficult.

SEDC was enhanced to store the sensor data in the Power Management Database (PMDB) from the current flat file structure. The PMDB already serves as storage of power and energy related data for the power management software. Keeping SEDC sensor data and power management data in the same PMDB provides unified access to all system sensor data.

Both SEDC data holding tables are partitioned, which is supported in the PostgreSQL. The table partitions prevent tables to grow too large, improve performance and help with data archiving. The 'cc_sedc_data' tables hold cabinet level collected sensor values. The 'bc_sedc_data' tables hold blade level collected sensor values.

The HSS Power Management daemon, *xtpmd,* supports table partitioning in its database maintenance via the *xtpmdbconfig* command line tool. The purpose of this tool is to configure the maximum age of the data, which is stored in the database before being rotated out. Additionally, it is used to configure a user defined hook utility to execute when certain table partitioning events occur.

To view the SEDC data collected at cabinet and blade levels via database queries as follows:

*SELECT value FROM pmdb.cc_sedc_data WHERE sedc_id=SEDC_CC_T_MCU_TEMP*

To view the sensor specifications details via database queries as follows:

*SELECT value FROM pmdb.bc_sedc_data WHERE sedc_id=SEDC_BC_T_PCB_TEMP*

## X. HSS Controller Monitoring

The Controller Vitality Check (CVC) daemon, *cvcd*, executes on all of the blade and cabinet processors (Intel EP80579 Tolapai) in the Cray XC system and is used to monitor the HSS blade and cabinet controllers. Initially the CVC daemon only monitored the controller processor load and memory usage.

There are three levels of resource utilization that are monitored. An "INFO" watermark (e.g. memory consumption eclipsing 50%), a "WARNING" watermark (e.g. memory consumption eclipsing 75%), and an "ERROR" watermark (e.g. memory consumption eclipsing 90%). A syslog entry is made when a measurement goes above or below the defined watermarks for each resource. These syslogs are forwarded to the SMW for later analysis.

Exceeding the INFO watermark means that a condition is unusual but the system administrator shouldn't be overly worried about it. Exceeding the WARN watermark means that the condition denotes something abnormal and more serious. Exceeding the ERROR watermark should never happen and is indicative of something going very wrong and that the controller could fail at any moment.

The CVC daemon implements a pluggable architecture. Three additional plugins have been added as follows:

1. **Tolapai Memory Errors:** A CVC plugin has been added to detect and handle memory errors (recovered and unrecovered) on the Tolapai processor.
2. **Tolapai PCIe Advanced Error Reporting (AER) Errors:** The Tolapai Linux supports PCIe Advanced Error Reporting (AER). This mechanism is used to report PCIe errors that are seen on the Tolapai from HSS embedded devices on the board.
3. **Controller Kernel Oops Errors:** There are times when the code in the kernel will have errors and result in a Linux controller fault. The new plugin monitors the kernel logs for kernel oops errors.

The SMW command, *xtdaemonconfig*, is used to configure the CVC plugins. If any of these plugins detect an error an HSS health event is generated. SEC is used to send an email to the administrator alerting them that an uncorrectable memory error has occurred on the controller.

## XI. System Notificatiom

Cray has standardized on utilizing the Simple Event Correlator (SEC) for alerts and alarms within Hardware Supervisory System (HSS). HSS alerts and alarms are processed by SEC and trigger the appropriate rule. The site can customize each rule or add additional rules as needed.

The HSS software implements notification utilizing the Open Source Simple Event Correlator (SEC) package, sec-2.7.0, and an SEC support package, cray-sec-4.0.0. The SEC support package contains control scripts to manage the starting and stopping of SEC around a Cray XC boot session, in addition to other utilities. The SEC support package contains a rule set that is based on Cray Service personnel best practices.

The HSS SEC package also includes the *check_xt* utility, which reports on state changes in the Cray system (such as system boots or compute nodes going down), and logs data about the state of nodes and jobs. *check_xt* is designed to work in conjunction with SEC to send email and text alerts about critical system issues.

SEC supports individual instances of SEC per partition. There is no limit on the number of partitions. Each partition supports different email recipients.

*xtbootsys* automatically starts SEC on both High Availability (HA) and non-HA systems.

There have also been a number of new rules included in SEC as follows:

- Detect excessive cabinet power draw
- Emergency cabinet power down
- Cabinet air temperature alert
- Cabinet water pressure alert
- Node memory errors
- Aries PCIe link change
- RDMA timeout
- Get ALPS Process ID (APID) on job failures

See the Cray publication, Configuring SEC Software for a Cray XC, Cray XE, or Cray XK System, S-2542-7202 for more details [6].

## XII. Cray Linux Environment (CLE) Enhancements

The Cray Data Virtualization Service (Cray DVS) is a distributed network service that provides transparent access to file systems residing on the service I/O nodes and remote servers in the data center. Cray DVS provides a service analogous to NFS.

Cray DVS provides I/O performance and scalability to a large numbers of nodes, far beyond the typical number of clients supported by a single NFS server. Impact on compute node memory resources, as well as operating system noise, is minimized in the Cray DVS configuration.

There have been a number of diagnosability enhancements made to Cray DVS as follows:

- Added /proc based mechanism that can be used to identify hung Cray DVS request processes.
- Improved Cray DVS log messages by including the file name or path in many of the file system error messages. Also provided extra logging information for jobs that have to be killed due to a server failure.
- Changed Cray DVS log timestamps from jiffies to seconds to improve readability of the logs.
- Added support to Cray DVS to periodically sync file system data in order to minimize the number of applications that are killed when a DVS server crashes.
- Improved the Cray DVS error recovery, logging, and failover support when I/O request responses fail to arrive.
- Added statistics to track the periodic sync when a file is closed. Added two new fields to sync_stats: close syncs and closing time.

There are also a number of enhancements for ESTALE *errno* (Stale NFS file handle) error handling in Cray DVS including:

1. **Periodic sync**: Using the new periodic sync capability, Cray DVS marks a file for syncing when writes to that file have subsided and a tunable time component had expired.
2. **Close/Re-Open**: If a file has never been written to or has been synced and encounters an ESTALE error, Cray DVS closes the file and re-opens it based on a tunable time-based retry component.
3. **Different Server**: Finally the Cray DVS client chooses a different server using a similar approach for server failover (i.e. all clients choose the next available DVS server and subsequent file operations are directed to that server).

By default, this feature is in the CLE 5.2UP02 release, but can be turned off by setting the value in /proc/fs/dvs/estale_max_retry to zero (0).

## XIII. Future Considerations

Additional work is planned to continue to enhance system diagnosability. One area is to include additional workload tests that have been found to test and stress the Cray XC system in ways that diagnostics have not been able to. While workload tests have been shown to expose problems, they generally are not good diagnostic tools. Often times they can only report that "something" has happened that was not expected.

Diagnostics need to continue to evolve and be enhanced to not just test individual components, but to also test the system in similar ways to the workload tests. Future work within diagnostics will focus on understanding how the workload tests execute and stress the Cray XC system and to then enhance the Cray XC system level diagnostics.

There are plans to provide additional resiliency, monitoring and logging within the CLE software stack including both DVS and Lustre.

As with any large data problem, Cray is working to provide diagnostic data analysis tools that can sift through the data collected on the SMW and make the connections between the data and failures or potential failures on the Cray XC system. Analysis tools can quickly review data over a period of time and present the data of interest to the system administrator.

Finally the data can be presented in an HSS dashboard where operations staff can quickly see where failures are being detected, alerts and alarms, and monitor data within OpenStack.

## XIV. Summary

Cray has provided a number of diagnostics, commands, and utilities that enhance the system diagnosability of the Cray XC system. The focus of system diagnosability has been on ensuring that each component is functioning properly by ensuring that each component can be validated and that all data is captured. Each aspect of the tool chain has been enhanced on the Cray XC system.

Enhancements have included BIOS, SMW commands, utilities, and diagnostics, as well as, power and thermal data and event logs. Future enhancements are planned to continue to improve system diagnosability of both the hardware platform and software components of the Cray XC system.

## References

[1] Nvidia Healthmon, DU-06718-001_vR331, March Best Practices and User Guide, March 2014, http://docs.nvidia.com/deploy/healthmon-user-guide/index.html.

[2] XDD User Guide, Version 7.0.0.rc28, February 2015, https://github.com/ORNL/xdd.

[3] High Performance Computing Challenge (HPCC) Benchmark, Version 1.4.2, http://icl.cs.utk.edu/hpcc/.

[4] High Performance Linpack (HPL) Benchmark, Version 2.1, http://www.netlib.org/benchmark/hpl/.

[5] Intel MPI Benchmark, Version 3.2.4, https://software.intel.com/en-us/articles/intel-mpi-benchmarks.

[6] Configuring SEC Software for a Cray XC, Cray XE, or Cray XK System, S-2542-7202.